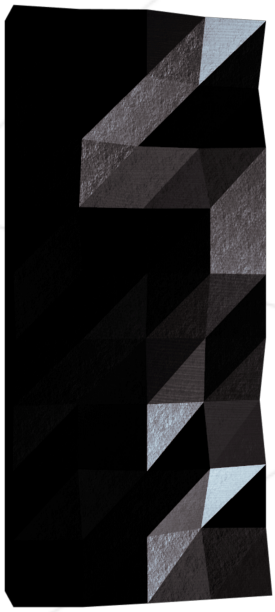


Training Piscine datascience - 4

The futur

Summary: Today, you will see some model to predict the future

Version: 1.00

Contents

Chapter I

General rules

- You have to render your modules from a computer in the cluster either using a virtual machine:
    - You can choose the operating system to use for your virtual machine
    - Your virtual machine must have all the necessary software to realize your project. This software must be configured and installed.
- Or you can use the computer directly in case the tools are available.
    - Make sure you have the space on your session to install what you need for all the modules (use the goinfre if your campus has it)
    - You must have everything installed before the evaluations
- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a `0` during the evaluation.
- We encourage you to create test programs for your project even though this work **won't have to be submitted and won't be graded**. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.
- By Odin, by Thor ! Use your brain !!!

Chapter II

Introductions

Data Scientists often use tools such as techno python, Jupyter Notebook, Julia ...

It is up to you to find the tools that suit you. You are free to use any language of your choice for this module.

The role of the data scientist is to predict "the future" with automatic learning models on past data, he must be a force of proposal to explain the possible interest to the implementation of his models, create tools to help decision making.

Chapter III

Exercise 00

	Exercise 00
Exercice 00 : Confusion Matrix	
Turn-in directory : $ex00/$	
Files to turn in : `Confusion_Matrix.*`	
Allowed functions : `lib to display the image only`	

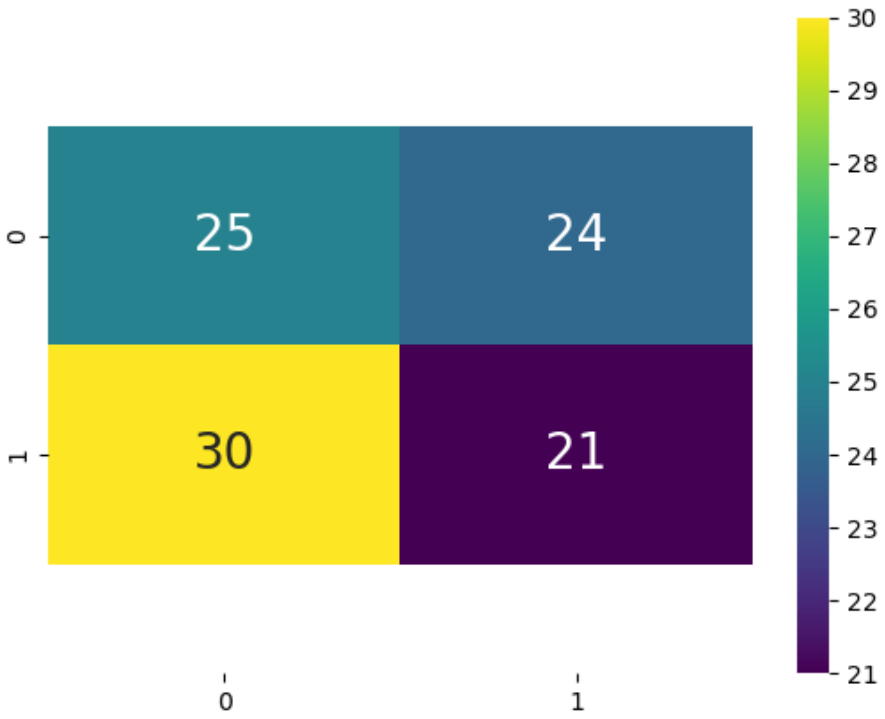
- You have to do the calculations by yourself. But you can use any library to display the graph
- Print and display matrix confusion
- You will have to re-use this exercice later on, so make sure you understand it. This will be checked during the evaluation

```
$>./Confusion\_Matrix.* predictions.txt truth.txt
     precision   recall f1-score    total

Jedi      0.45     0.51     0.48       49
Sith      0.47     0.41     0.44       51

accuracy                    0.46      100

[[25 24]
 [30 21]]
 $>
```



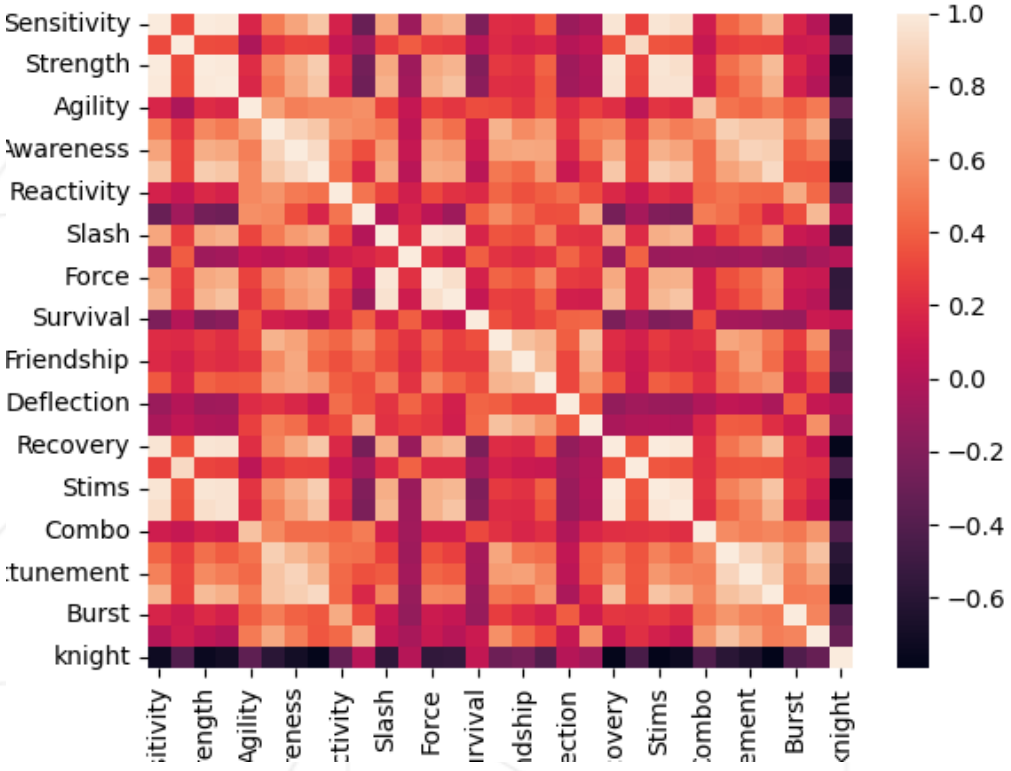
If this exercise is wrong, the eval stop here : it's over.

Chapter IV

Exercise 01

	Exercise 01
Exercice 01 : It is warm	
Turn-in directory : $ex01/$	
Files to turn in : `Heatmap.*`	
Allowed functions : `All`	

- Make a Heatmap to see the Correlation Coefficient between the data



Chapter V

Exercise 02

	Exercise 02
Exercice 02 : Variances	
Turn-in directory : $ex02/$	
Files to turn in : `variances.*`	
Allowed functions : `All`	

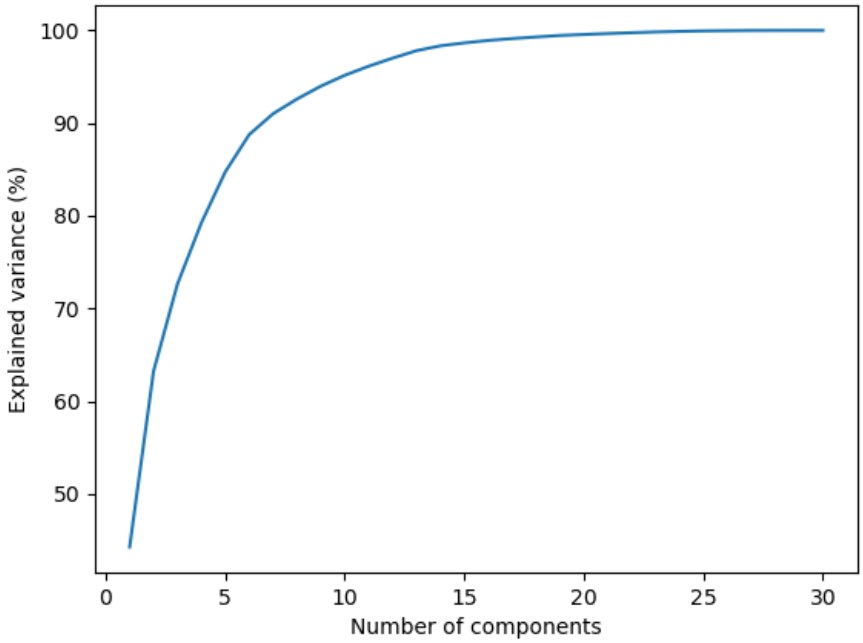
- Calculate the variance of each skill
- Add up the variances to see how many components needed to reach 90
- Display a graph representing the addition of your variances

For Exemple:

```
Variances (Percentage):
[4.48960353e+01 1.84721038e+01 9.18338543e+00 6.44633272e+00
 5.35186638e+00 3.89518676e+00 2.20877147e+00 1.56404980e+00
 1.34482203e+00 1.13191496e+00 9.83404501e-01 9.38664017e-01
 8.41969101e-01 6.84759840e-01 4.79278013e-01 2.84395361e-01
 2.57613460e-01 1.90436550e-01 1.66006155e-01 1.55535318e-01
 9.90525123e-02 9.52472611e-02 8.82086046e-02 7.83341693e-02
 5.66200463e-02 4.99139931e-02 2.59875603e-02 2.21604256e-02
 5.12408800e-03 2.39251211e-03 4.27853873e-04]

Cumulative Variances (Percentage):
[ 44.89603531 63.36813909 72.55152452 78.99785725 84.34972363
  88.2449104  90.45368186 92.01773166 93.3625537  94.49446866
  95.47787316 96.41653717 97.25850628 97.94326612 98.42254413
  98.70693949 98.96455295 99.1549895  99.32099566 99.47653097
  99.57558349 99.67083075 99.75903935 99.83737352 99.89399357
  99.94390756 99.96989512 99.99205555 99.99717963 99.99957215
 100.        ]
```

For Exemple:



Perhaps you should rework your data with what you have seen in the previous module if your answers are too far from the examples.

Chapter VI

Exercise 03

	Exercise 03
Exercice 03 : Feature Selection	
Turn-in directory : $ex03/$	
Files to turn in : `Feature_Selection.*`	
Allowed functions : `All`	

The data seems too Multicollinearity, you will have to make a Detecting Multicollinearity model.

There are a number of models for detecting the importance of variables (Lasso, Backward Elimination, Step Forward Selection, ...)

But here you will have to use the Variance Inflation Factor (VIF)

- Display the VIF of your data
- Keep only the features so that the VIF goes under 5, and display the features

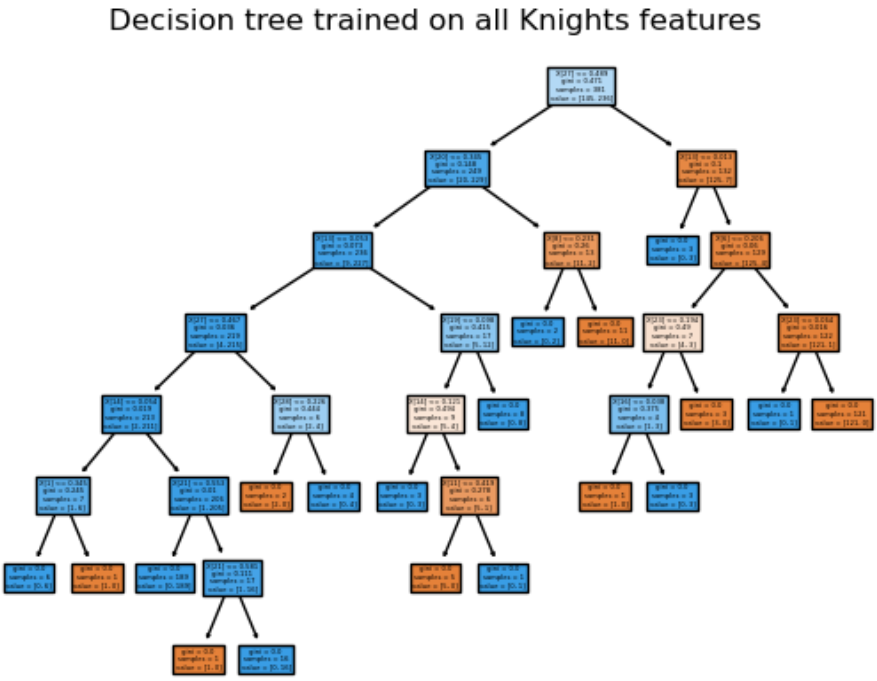
```
                   VIF  Tolerance
Empowered      36.763714  0.027201
Stims         405.023336  0.002469
Prescience     60.041733  0.016655
Recovery      799.105946  0.001251
Strength     3786.400419  0.000264
...                  ...       ...
```

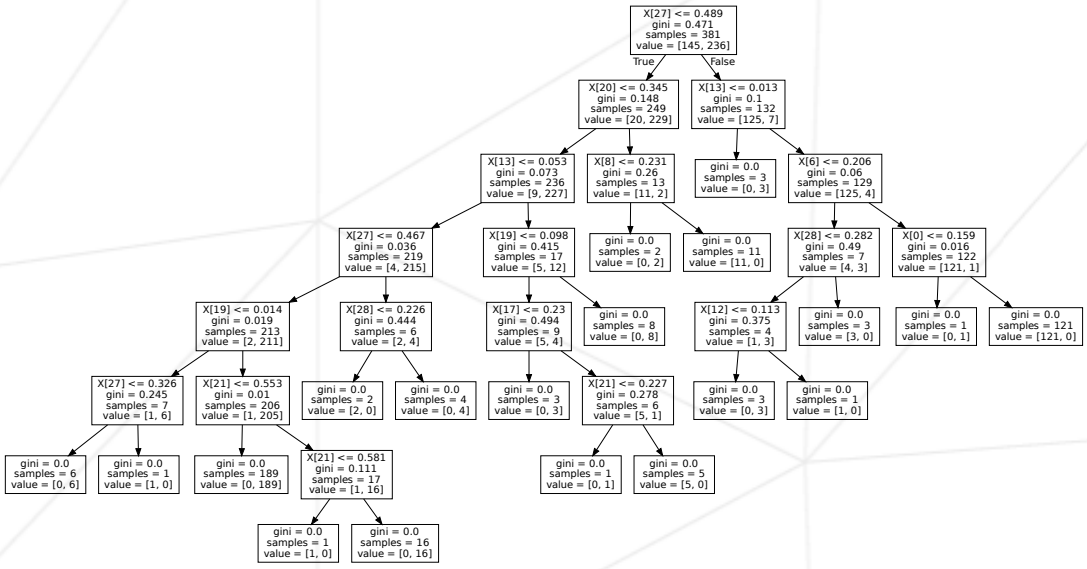
Chapter VII

Exercise 04

	Exercise 04
Exercice 04 : Forest	
Turn-in directory : $ex04/$	
Files to turn in : `Tree.*`	
Allowed functions : `All`	

- Make a Decision Tree Classifier or Random Forest Classifier model
- Display the tree in a graph
- Your program must take the Train_knight.csv file as first argument, the Test_knight.csv file as second argument and write a Tree.txt file with the prediction (one prediction per line, either Jedi or Sith, in the same format as the prediction.txt file)
- Your model must have minimum 90% of f1-score





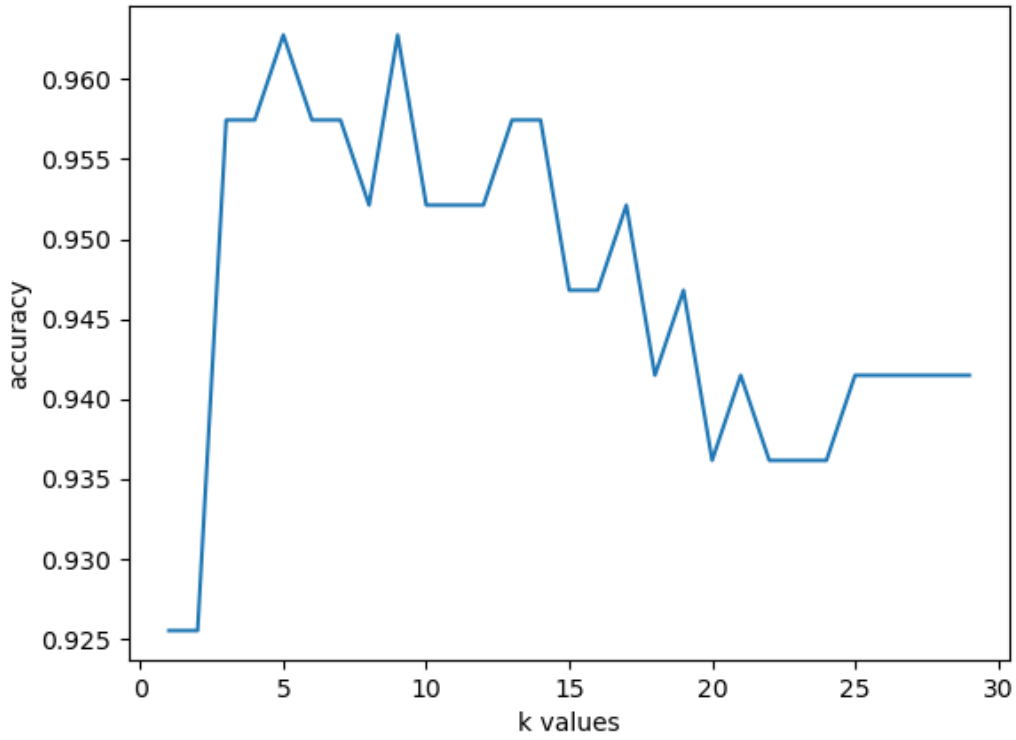
You will probably need your files Training_knight.csv and Validation_knight.csv from the module03

Chapter VIII

Exercise 05

	Exercise 05
Exercice 05 : KNN	
Turn-in directory : $ex05/$	
Files to turn in : `KNN.*`	
Allowed functions : `All`	

- Your program must take the Train_knight.csv file as first argument, the Test_knight.csv file as second argument and write a KNN.txt file with the prediction (one prediction per line, either Jedi or Sith, in the same format as the prediction.txt file)
- Make a KNN that calculates the precision % according to the number of k-value in your Validation
- Display the graph
- You must have a minimum of 92% f1-score



You will probably need your files Training_knight.csv and Validation_knight.csv from the module03

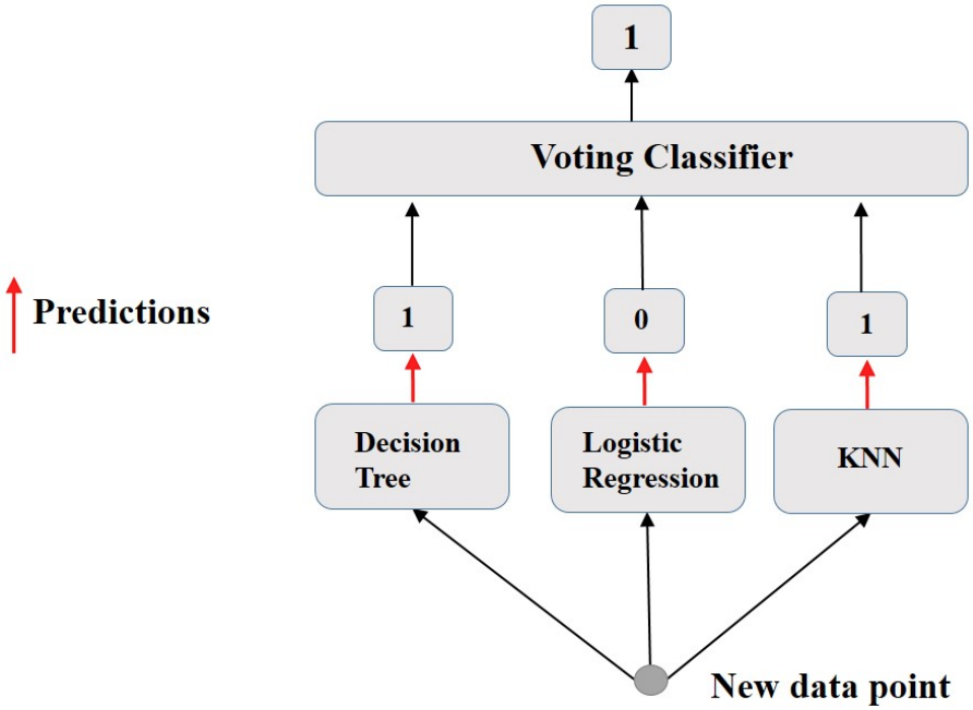
Do you have convergence problems? If so, you should use what you learned in the previous exercises in this module and in the Predecent module to improve your predictions.

Chapter IX

Exercise 06

	Exercise 06
Exercice 06 : democracy !	
Turn-in directory : $ex06/$	
Files to turn in : `democracy.*`	
Allowed functions : `All`	

- Choose a third model of your choice
- Make a Voting classifier
- Your program must take the Train_knight.csv file as first argument, the Test_knight.csv file as second argument and write a Voting.txt file with the prediction (one prediction per line, either Jedi or Sith, in the same format as the prediction.txt file)
- You must have a minimum of 94% f1-score



You will probably need your files Training_knight.csv and Validation_knight.csv from the module03

Chapter X

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.

`The evaluation process will happen on the computer of the evaluated group.`